PRE-PRINT VERSION

Bauer, Christine (2014). A framework for conceptualizing context for intelligent systems (CCFIS). Journal of Ambient Intelligence and Smart Environments, 6(4), IOS Press, pp 403-417. DOI: 10.3233/AIS-140269

The final publication is available at IOS Press through http://dx.doi.org/10.3233/AIS-140269

The author is now affiliated with:
Department of Computational Perception, Johannes Kepler University Linz,
Altenberger Strasse 69, 4040 Linz, Austria, E-mail: christine.bauer@jku.at, Web: http://christinebauer.eu

A framework for conceptualizing context for intelligent systems (CCFIS)

Christine Bauer[a,*]
[a]*Department of Information Systems & Operations, Vienna University of Economics and Business, Welthandelsplatz 1, D2, 1020 Vienna, Austria*

Abstract. As sensing technologies advance, designers face an increasing variety of exploitable context when they create intelligent context-adaptive systems. In this opaque conglomerate of context, designers of intelligent systems find it difficult to select the elements that most effectively help a system tap into its full potential of intelligence. In emerging technology-driven areas, there is a vital need for a universally valid, flexible structure that provides the basis for target-oriented research using a shared conceptualization. In fact, such a framework is essential to enable, yield, and foster sustainability in a novel and interdisciplinary research field. For this reason, this paper introduces a cohesive and flexible conceptual framework for conceptualizing context for intelligent systems (CCFIS). Based on an example of the pervasive advertising domain, this paper shows how designers can conceptualize context in adherence to CCFIS.

Keywords: Framework, conceptualization, context, context-adaptive systems, intelligent systems

1. Introduction

In 1991, Weiser [56] presented the vision of a world where technology is smoothly woven into people's everyday lives. This vision reflects intelligent environments [18], in which applications are aware of their context [17] and can change their behavior according to this context [7].

Early research in context-aware computing goes back to the 1990's [46]. Since then, researchers have studied context-aware systems from myriad angles, such as to advance sensor technologies [e.g., 41] and the processing and storing of context data [e.g., 18].

However, research on intelligent context-adaptive systems is scattered and mainly prototype-driven. In the field, the research community works on individual problems that need to be solved in the different phases of system development (e.g., data collection challenges, data aggregation, sensor fusion, adaptivity, machine learning, etc.). Probably due to the versatility of challenges that researchers face in the field, the community has not yet

[*]Corresponding author. E-mail: chris.bauer@wu.ac.at.

established a holistic and systematic methodology that specifies how intelligent systems should be built.

A pivotal ingredient of intelligent context-adaptive systems is known as 'context'. Despite the importance of context for intelligent systems, there still lacks a single, unified definition of it [12], and works that attempt to enumerate context elements are largely divergent [2]. A *context element* describes one contextual aspect of the variety of context. Examples for context elements include the location of a user, outdoor temperature, speed of a car, etc.

However, when system designers plan intelligent context-adaptive systems, they need to anticipate the relevant combinations and characteristics of context elements that could occur in a real situation before the system is actually implemented. Given the vast number of context elements available to consider in intelligent systems, anticipating all relevant elements for a particular system at the design stage is a difficult task. Thereby, the choice is not whether to conceptualize context (and, thus, to simplify, reduce, and objectify it) but *how* to do it [11]. In other words, system designers need a systematic approach for identifying potentially relevant context elements. Yet, in the absence of such an approach, system designers currently seem to rely only on their personal knowledge, intuition, experience, and cognitive abilities.

To address these challenges, the present paper proposes a framework for conceptualizing context for intelligent systems (CCFIS). Specifically, this paper targets researchers and system designers in the field of intelligent context-adaptive systems. Those researchers and designers can use the suggested CCFIS as a guide for eliciting potentially relevant context elements for a certain, envisaged intelligent context-adaptive system. CCFIS is designed to cover the *conceptualization of context* as an additional task in the development process, which designers do not have to consider when developing systems that are not context-adaptive. Accordingly, the proposed framework is designed to cover the conceptualization of context in the creativity phase of system design, which is more or less explicitly integrated in all software development process models. Note that the framework is conceptual and is not intended to replace formal methodologies for context modeling, nor does the framework claim to represent a formal method.

The article is structured as follows: The next section introduces the reader to the field of context conceptualization. The third section documents how the framework was built. The fourth section proposes a framework for deriving the relevant context elements for an intelligent context-adaptive system (CCFIS). The fifth section discusses an instantiation in the field of context-adaptive advertising displays (digital signage, pervasive advertising) in retail. A critical discussion of the findings follows in the sixth section, and the paper closes with a conclusion.

2. Conceptual foundations

2.1. Recent conceptions of context awareness

In its early days, context awareness built on the assumption of a single-user, single-device relationship. Accordingly, conceptions of context awareness considered context elements (situational factors) that influence the interaction between one application and its single user [14]. Because devices had limited sensing capabilities at that time, the context elements that could be adopted by 'intelligent' systems were largely confined to the user's location and time [2].

Recently, there was a momentous shift from context awareness towards "socially-aware" concepts [34]: Because of advanced technology, multiple users and multiple devices may interact in heterogeneous environments [34]. This novel conception is enabled by advancements in socially-aware computing, especially reality mining, which allows detailed insights into human life [42]. Beyond built-in sensors in devices (such as the accelerometer, compass, and gyroscope that are nowadays embedded in any common smartphone), sensory input can also be gathered from any kind of information source (e.g., online sensors such as Twitter feeds or playback sensors that generate data streams from repositories) [35]. In short, almost anything can be used as sensory input; the number and versatility of exploitable context elements for intelligent systems have increased enormously. As a result, the need for a systematic conceptualization of context is more vital than ever before.

2.2. Approaches to conceptualize context

In the field of software engineering, many publications are dedicated to the conceptualization of some issue (e.g., software testing as a service [57], the software life cycle [1], interpersonal relationships in agile information systems development [37]), which forms a basis for further software engineering efforts.

In addition to conceptualizing such issues, system designers must also conceptualize *context* to seamlessly support the intelligence of all conceivable behavior of an intelligent context-adaptive system. "How are dimensions of context identified, quantified, and interrelated for each situational purpose?" [6]. Bauer and Spiekermann [3] define context conceptualization as "*the process* by which a personalization situation is deconstructed into measurable and logically disjunctive information units, all of which must be combined to create an adaptive service."

Scholars already work on deconstructing and identifying disjunctive information units; they propose so-called *generic context models*. While a *specific context model* specifies relevant context for a certain context-adaptive system, a *generic context model* aims to describe the variety of context independent of any specific system. This approach to context conceptualization (context modeling) is static.

Various generic context models exist, and these differ considerably in the variety of context categories that they include. While some models organize context into only a few categories and hierarchy levels [e.g., 33], others provide a rich array of context categories and examples [e.g., 50]. However, the richer models are not necessarily comprehensive: Bauer [2] could show that 27% of the context elements reported in the Pervasive Computing Magazine are not covered by any of the 13 models considered in the study. Additionally, although many models contain some identical context elements (e.g., location, time, environment), the majority identifies some unique context categories (cf. Tab. 1). This heterogeneity of context categories also indicates that the models rarely build on each other.

Overall, existing generic context models organize the variety of context and, thus, provide a structured description of what context is about on a more or less abstract level. However, conceptualizing context requires more than a static structure with predefined categories. We claim that designers can only identify meaningful, relevant context elements (from the variety of context) for a specific intelligent context-adaptive system by using a *process* approach.

Against this background, the present paper suggests a framework for conceptualizing context that considers a process-orientation instead of a static approach. In doing so, the framework builds on generic context models – thus, building on previous findings – and combines them with procedural techniques.

2.3. Integrating context conceptualization into the software development process

Traditional software engineering process models following the software development life cycle (SDLC) [55], such as the waterfall model [44], spiral model [4], or V-model XT [8], have long development cycles with stable requirements in one cycle. Agile software development methods – including extreme programming (XP) [31] – dynamically adapt requirements in frequently rotating cycles.

Still, both approaches to software development do not explicitly consider the complexity, variety, or multi-dimensionality of context. We suggest that, irrespective of the dynamism of requirement changes, defining context requirements is an *additional*, but indispensible, task *on top of* determining other system requirements. System designers must identify the context that best supports the system's adaptivity goal. As Chalmers [10] points out, the question faced by system designers is not whether to reduce, objectify, or constrain context for a context-adaptive system, but how to do it.

Identifying the context elements that serve the purposes of a particular intelligent context-adaptive system is complex [13, 19], not only due to the versatility of context, but also because of the challenge accompanying implicit interaction between user and system. At present, the development of context-adaptive systems is governed by ad-hoc processes [49] and "wild-west" prototyping [36], and system designers seem to use a so-called "i-methodology" [40].

3. Framework development

Developing a framework is an inherently iterative search and design process that follows a set of continuous cycles of generating and evaluating design alternatives. By taking an *informed argument approach* [23], we iteratively used information from the knowledge base throughout the design process of creating CCFIS as it is presented in Section 4. In the initial phase we tapped the knowledge base of the research domain 'context conceptualization' and critically analyzed literature in group reflection sessions. Then, we gradually improved a CCFIS draft

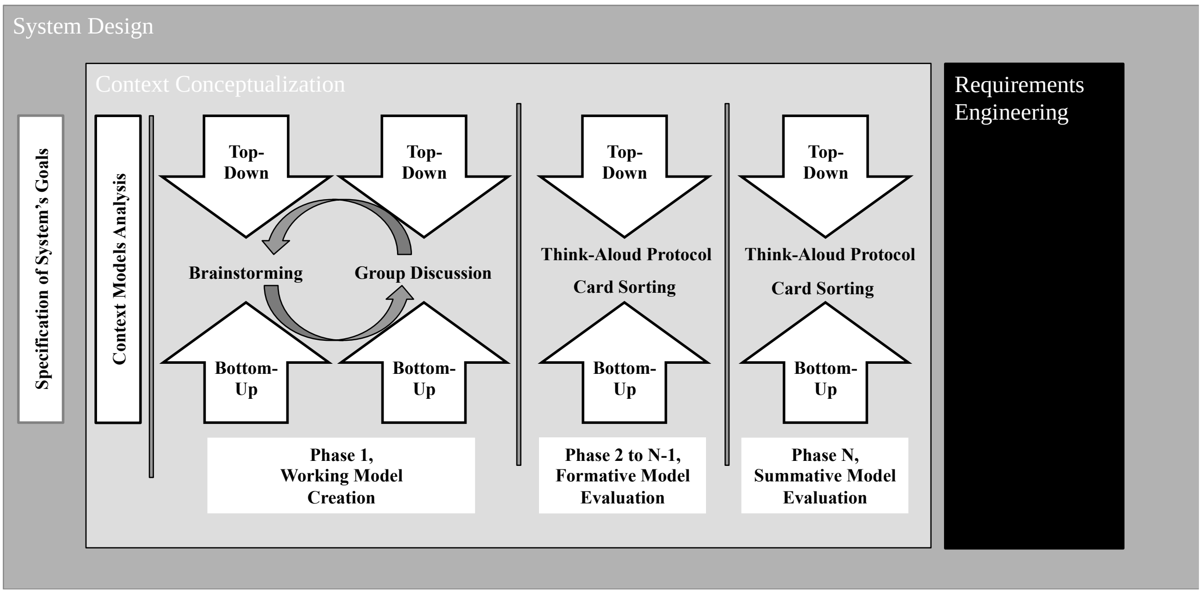

Fig. 2. A framework for conceptualizing context for intelligent systems (CCFIS)

version by following the informed argument approach and generating enhanced design alternatives. Group reflection phases after each generate/evaluation cycle helped to converge different design alternatives into one coherent framework. Each cycle contributed to the advancement of CCFIS, leading to the final version as presented in Section 4. This final version was then applied in a case study for a summative evaluation of the framework. The approach of generating and evaluating CCFIS is visualized in Fig. 1.

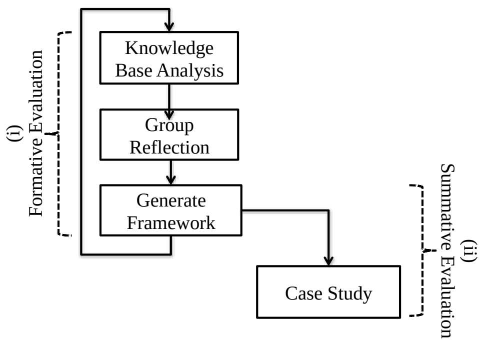

Fig. 1. The generation/evaluation cycle

4. The framework for conceptualizing context

The purpose of the proposed CCFIS is to conceptualize context for a specific intelligent context-adaptive system; i.e., to identify the context elements (from the variety of context) that are relevant for the specific system. As, beyond determining other system requirements, designers must identify the context elements that best support the system's adaptivity goal, they shall determine relevant context elements independently of any considerations regarding the deployable technology (paradigm of opportunistic sensing [35]). Thus, system designers apply CCFIS in the creativity phase of system design, no matter which software development approach is generally followed. The independence from any technical implementation is crucial in the early stage of system design; the question should not be what technical implementation to use, but which purpose the system should serve. Work on the "multi-modal, multi-model approach" shows that anticipating technical implementations may ex-ante restrict design options for a system and may even prevent a system from best fulfilling its purpose [16, 32].

Basically, there are five main stakeholders in an intelligent context-adaptive system: (1) a system designer who designs a system, (2) a data provider (licenser) who provides data and may license these to clients, (3) an integrator who integrates the system and data provision and provides a full package to clients, (4) a system operator who provides and operates a system, and (5) a user who uses the respective system, either implicitly or explicitly. In many cases the system operator, who provides and operates the system, will have more or less stake in it than the user, who uses it.

Initially, CCFIS takes the *system operator's perspective*. Simultaneously, it integrates the *user's perspective*, since it is important to understand potential users' preferences, habits, desires, behavioral patters, etc. As a result, CCFIS' approach to conceptualizing context is both top-down and bottom-up. The process is top-down driven (model driven) to characterize the field of application on behalf of the system operator. Bottom-up, it is driven by the aim to understand the user. CCFIS takes and considers the system operator's and the user's perspective, but it addresses the system designer, who is the actor that will apply CCFIS.

As indicated above, CCFIS' approach to conceptualizing context is both top-down and bottom-up (Fig. 2). The top-down approach is informed by an analysis of generic context models and involves reflecting on context categories that might be relevant for the envisaged system. In contrast, the bottom-up approach considers context elements in each of the identified context categories. System designers use these two perspectives to analyze the system's context in up to N phases, refining the analysis each time.

Applying CCFIS supports designers in determining the context elements that may be relevant for a given intelligent context-adaptive system or bundle of such systems. Finally, the identified, relevant context elements are represented by an ontology.

4.1. Specification of the envisaged intelligent system's goals

Before system designers may conceptualize context, they must specify the envisaged system's goals and functions at a business logic level. System designers need this specificity in order to identify the context that may support or enable the system [13].

4.2. Context models analysis

The first step is to analyze generic context models from the multidisciplinary domain of context (including computer science, psychology, and business) in order to identify a fitting one for the envisaged system.

An overview of various published context models is given in Tab. 1. Researchers and system designers should be able to draw from the list quickly, instead of having to perform a thorough literature review every time they design a new intelligent system.

From the pool of existing generic context models, system designers should select the one that appears most appropriate for the envisaged intelligent system.

We suggest that researchers/designers use an open card sorting procedure. However, they can use other techniques as well. For the card sort, each generic context model is written/drawn on a card together with a concise representation of the context elements of the respective model. Then, the participants sort these cards into piles according to their perceived fitness for the envisaged intelligent system.

Besides the 'paper and pencil' approach, teams can apply electronic card sorting, which eases both the sorting task and data collection. To come up with a jointly accepted hierarchy of models, Maier and Stix [38] propose an automatic construction algorithm.

4.3. Working model creation

4.3.1. Top-down conceptualization

Researchers/system designers begin the first phase of conceptualizing context by challenging the applicability of the context model they selected for the envisioned intelligent context-adaptive system. As teams participate in top-down brainstorming and group discussions, they will a) *rename* categories, b) *refine* the model, and/or c) *extend* the model to embrace further context elements that may also be domain-specific.

The model should also reflect the entity providing the intelligent system. To acknowledge this additional stakeholder, we suggest adding a high-level category called 'domain-specific context' to the working model.

As they work on the second level of model detail, researchers/designers should again rename, refine, and extend context categories.

4.3.2. Bottom-up conceptualization

When teams refine and extend a generic context model, they do not yet produce a sufficiently thorough conceptualization of context. To specify "context feature space on the third level" [48], we need to consider context from the user's situational perspective. Thus, researchers/designers must think about the specific situation in which the intelligent system is provided and try to bring together all context elements that could be needed to support adaptivity in the end.

Researchers/designers discussing the specific context elements needed for the intelligent system will find it fruitful to apply scenario-based design [9]. The conceptualizing team should think about *situational scenarios* involving the adaptive functionality provided by the envisioned intelligent system.

The gap between the situational detail and the broader categories identified in the top-down conceptualization requires further structuring to allow for a thorough conceptualization of context.

Therefore, CCFIS introduces a hierarchical specification of the top-down categories on three levels: a macro, micro, and situational level. The macro level is valid for all model applications. It should be considered as a further refinement of the categories as identified in the top-down conceptualization, but specific to the envisioned intelligent system use.

Tab. 1. Overview of generic context models, sorted by numbers of hierarchy levels and nodes

Authors	Type of publication	H *	N *	Context elements
Prekop and Burnett [43]	Journal	1	2	agent, activity
Lucas [33]	Journal	1	3	physical, device, information context
Lieberman and Selker [30]	Journal	1	4	user, physical, computational environments, interaction history (time)
Bradley and Dunlop [6]	Journal	1	5	task, physical, social, temporal, cognitive
Turel [54]	Journal	1	5	who (user), when (time), where (environment), why (motivation), what (task/goals)
Schmidt, Aidoo, Takaluoma, Tuomela, Laerhoven and Velde [47]	Lecture Notes	2	3	self (device state, physiological, cognitive), environment (physical, social), activity (behavior, task)
Dix, Rodden, Davies, Trevor, Friday and Palfreyman [15]	Journal	2	4	infrastructure, system, domain, physical context
ISO 9241-210:2010 12 15 [25]	Standard	2	5	users or groups of users (knowledge, skill, experience, education, training, physical attributes, habits, preferences, capabilities), goals and tasks, technical (hardware, software, materials), physical (thermal conditions, lighting, spatial layout, furniture), social and cultural environment (work practices, organizational practices, attitudes)
Jameson [26]	Journal	2	9	situation's behavior, consequences for user (e.g., interestingness), utility for user, features of the situation (e.g., user's location), current state of user (e.g., cognitive load), longer-term properties of user (e.g., knowledge, interests), readings from context sensors (e.g., GPS), readings from physiological sensors, user's behavior with the situation
Jumisko-Pyykkö and Vainio [27]	Journal	2	6	social (persons present, interpersonal actions, culture), physical (spatial location, functional place and space, sensed environmental attributes, movements and mobility, artefacts), technical and information (other systems and services, interoperability, informational artifacts and access, mixed reality), temporal (duration, time of day/weeks/year, before/during/after, actions in relation to time, synchronism), task context (multitasking, interruptions, task type), context properties (level of magnitude, dynamism, pattern, typical combinations)
Lee, Kim and Kim [29]	Journal	3	2	personal (emotion, time, movement), environmental (physical, social)
ISO 9241-11:1998(E) [24]	Standard	3	6	users (user types, personal attributes), tasks, equipment (basic description, specification), organizational (structure, attitudes and culture, job design), technical (configuration), physical environment (workplace conditions, workplace design, workplace safety)
Sigg, Haseloff and David [50]	Journal	3	6	location (geographical, relative), time (period, relative), activity (action, task), constitution (biological, mood), environment (physical, technological, equipment), identity (user, social, organizational)
Truillet [53]	Lecture Notes	4	2	human factors (user, social, environment, task), physical environment (conditions, infrastructure, location)
Schmidt, Beigl and Gellersen [48]	Journal	4	3	human factors (user, social, environment, task), physical environment (conditions, infrastructure, location), time

H… number of hierarchy levels; N… number of nodes on the root+1 level;
* We assume that a generic context model follows a tree structure with ‘context’ as the root element.

For example, if the task of a user is specified as 'shopping', then the location of the service is the region where an intelligent system is launched (e.g., London). The micro level then filters this macro level information category and helps apply it to a specific system environment (e.g., a specific store in Notting Hill that has specific weather conditions, specific clientele, etc.). Accordingly, the specific system environment supplies more detailed information than the macro factors. Finally, the situational level describes an 'adaptive incident' or 'moment of service delivery' that happens in the system environment.

For context adaptivity, the situational level is the determining factor, as systems have to adapt to the actual conditions at the scene at the moment of service delivery. Still, understanding the micro and macro levels has proven useful for identifying the spectrum of relevant and available information sources.

4.4. Formative model evaluation

In the second phase of model development, the conceptualizing team evaluates the working model in a formative way by applying a systematic technique*: expert interviews* with *card sorting* [45]. Again, the team should choose both a top-down and a bottom-up approach. Note that the working model evaluation is formative in nature and is, thus, to be considered as part of the conceptualization efforts.

4.4.1. Formative top-down evaluation

The conceptualization team should represent the working model with cards (for instance, sticky notes). Participants should be briefly informed about the context of the project, the working model's first level of hierarchy (root+1) and the systematics of the macro, micro, and situational levels. When a participant does not understand a category's semantics, he or she should receive brief clarification. However, to avoid priming, no other questions should be answered.

Participants should be asked to interpret the overall working model while thinking-aloud so that their thoughts can be captured (top-down evaluation). They should be encouraged to rename categories or introduce new ones. Furthermore, they should be instructed to rearrange the categories until they feel satisfied with the model.

4.4.2. Formative bottom-up evaluation

In a second exercise, participants should be introduced to a specific situation in which the use of the intelligent system is envisioned. Then they should be asked to classify the context elements of the situation to their (rearranged) working model (bottom-up evaluation). Again, the participants should be given the opportunity to rename and rearrange the categories or introduce new ones until they feel that they can accurately classify the situation.

4.4.3. Iterations

The result of this two-sided approach of card sorting will probably encompass *rephrasing* and *rearranging* many of the categories on all three (macro, micro, and situational) levels and *introducing* new categories. The conceptualization team should repeat the formative model evaluation phase with additional participants until saturation, which means that the team does not gain any new insights from participants [5].

4.5. Summative model evaluation

When building taxonomies, a set of possible errors can be made [20]. Accordingly it is important to evaluate the resulting taxonomy. Taxonomies can be evaluated against many criteria: coverage of a particular domain, richness, complexity, granularity, consistency, completeness, conciseness, expandability, sensitiveness, etc. [20, 39]. For one source that shows how to deal with these criteria, see [21].

To ensure that the model is comprehensive and covers the particular domain of the envisaged intelligent system, the designer (team) should review the specification of the system's goals and verify that the model is a good fit. Assessment by humans against a set of criteria have, for instance, been used extensively in studies of ontologies and terminologies for systems in the context of biomedicine [e.g., 51, 52].

5. Case study: Application of CCFIS

This section demonstrates how CCFIS is applied. This paper's author provided the project specification (Section 5.1). Two researchers from the field of information systems (non-experts in the field of context-aware computing) formed the 'design team' for the case study and carried out the conceptualization efforts.

After presenting the specification of the envisaged system, this section explains how CCFIS is embedded in the software development process. Then, we describe how we conceptualized context for the pervasive advertising domain in retailing. We present the results in terms of a specific context model (Fig. 6). For details, see also [3].

5.1. Specification of the envisaged intelligent system's goals

The project owner's specifications to the design team can be summarized as follows: *The design team's task is to design an intelligent context-adaptive advertising system aimed at increasing the sales volume of ice cream. Networked digital displays (digital signage) are distributed between the outlets of a particular supermarket chain; they are used as interfaces to present context-adaptive advertising messages. The digital displays are mounted onto shelves or hang from ceilings. The supermarket offers standard and luxury ice cream. It is known that people are more responsive to ice cream advertisements when it is hot.*

We now outline the exemplary experience that a design team has when applying CCFIS. The main question to be answered by conceptualizing context for this setting is, which context elements are available and meaningful to allow an intelligent contextualization.

5.2. Integration of CCFIS within the process of intelligent system development

As this case study's main purpose was to evaluate the applicability of CCFIS, we opted for a simple software development process model, the Waterfall model. In doing so, we could begin straight away with conceptualizing context for the envisaged system after having read and understood the project specifications. Accordingly, we applied CCFIS in the early phase of requirement engineering.

We could have opted for a model that assumes that requirements engineering continues through the lifetime of a system (e.g., rational unified process, extreme programming, or SCRUM). We are aware that this simple Waterfall model is not widely applied in practice. Still, we believe that the chosen case study is illustrative for the integration of CCFIS in the software development process.

5.3. Context models analysis

Out of the set of generic context models presented in Tab. 1, we selected the context model by Schmidt, Beigl and Gellersen [48] as a starting point for the conceptualization process because it seemed to be the most appropriate one for the envisioned system.

We chose this model because of its structure, which seemed elaborate due to having several hierarchy levels. At the same time, this model offered only three nodes on level one, which seemed more flexible than, for instance, the nine nodes with the model by Jameson [22]. The design team considered flexibility to be important due to the advertising domain's versatility and complexity.

Fig. 3 provides an overview of this model by Schmidt, Beigl and Gellersen [48]. They distinguish context related to human factors from context related to the physical environment. On a second level, they operationalize human factors as information about users themselves, their social environment, and their tasks. The physical environment includes information about location, infrastructure, and physical conditions.

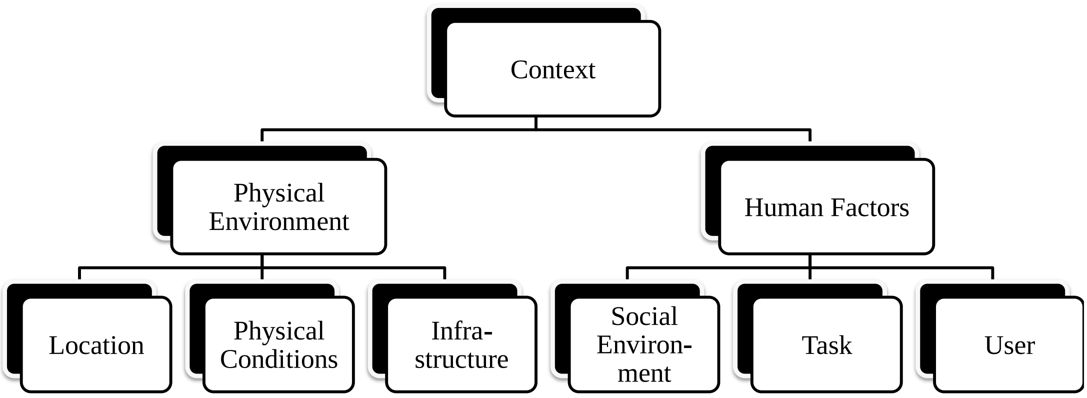

Fig. 3. The generic context model by Schmidt, Beigl and Gellersen [48]

5.4. Working model creation

After we selected the generic context model by Schmidt, Beigl and Gellersen [48] as a starting point, we challenged this model's applicability to the pervasive advertising domain.

5.4.1. Top-down conceptualization

Top-down brainstorming and group discussions led us to *refine* the model *for the advertising domain* and *extend* the model to *embrace all stakeholders*. More concretely, when we refined the top-level taxonomy, we renamed the 'human factors' category to 'consumer's environment'. The refined category refers to the particular role of humans that pervasive advertising targets: consumers. Furthermore, we opted for consumer 'environment' because this term is more precise than 'factors', particularly when embracing also social issues at lower levels of the taxonomy. We extended the model after considering that the entity delivering the advertising (the company advertising its products) should also be reflected. Accordingly, we added a high-level category called 'advertiser's environment'.

On the second level of model detail, we kept five of the six context categories as proposed by Schmidt, Beigl and Gellersen [48]: consumer profile, social environment, task, location, and conditions. We deemed it necessary, however, to rename and amend them to fit the advertising setting. Reflecting on the model and discussing it in the group led us to differentiate between 'manipulable' and 'non-manipulable' environmental conditions. 'Manipulable' conditions are environmental states that the system operator can influence (e.g., light conditions in a shop). 'Non-manipulable' conditions cannot be manipulated: they are given by nature (e.g., outdoor temperature). From the perspective of the system designer (who performs the context conceptualization), this distinction is significant because these two context categories entail different consequences for a system's operations and have to be considered adequately in system design. Non-manipulable environmental conditions are passively sensed and may or may not be used as input data for the system at hand. Manipulable environmental conditions, in contrast, can be actively controlled and designed in order to trigger a certain experience of interacting with the intelligent system. For example, while outdoor temperature is controlled by nature (non-manipulable), the temperature inside of a store could be set (manipulable) to a desired level. For advertising ice cream, for instance, when consumers are hot outside (non-manipulable) and hot inside (manipulable), they would probably buy more ice cream than if they were inside an air-conditioned store.

Furthermore, three additional context categories in the advertiser's environment category emerged on

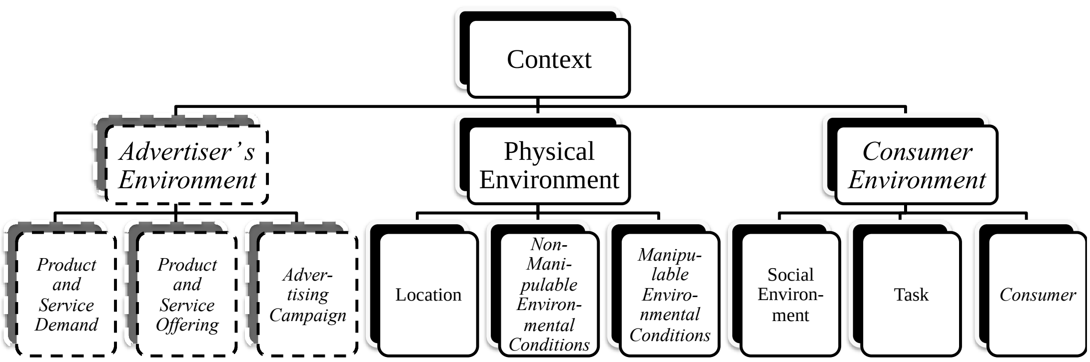

Fig. 4. Overview of the refinements of the context model for pervasive advertising [3]

the second level of model detail: product and service demand, product and service offering, and the advertising campaign itself, which is the object of the adaptive advertising system. Fig. 4 provides an overview of the model's refinements. Renamed elements are given in italics; additionally, a dashed line highlights newly introduced elements.

5.4.2. Bottom-up conceptualization

We followed a bottom-up approach, considering specific scenarios in which adapted advertisements are provided to a consumer. This approach contributed effectively to consolidating the various specific context categories needed to support such an advertising system.

Although the specification for the envisaged intelligent system was dedicated to the sale of ice cream, the design team concurred that the system has to be designed in a way such that it is scalable to cover advertising other products as well. Therefore, the team also considered several non-ice-cream-related scenarios.

One scenario focused, for instance, on a hair-coloring product designed to cover grey hair. Ideally, an advertisement for this product should be specific to a customer's sex, age, and hair color. Such an advertisement should only be publicly displayed when most people near the display are those the advertisement targets. It should be shown at the moment when the prospective customer is near hair-styling shelves. And it should only be shown if the product is on the shelf (or at least in stock).

In accordance with CCFIS, we introduced three hierarchical levels: a macro, micro and situational level. Fig. 5 provides an overview of the structuring approach, summarizing an excerpt of the final specific pervasive advertisement context model (context taxonomy).

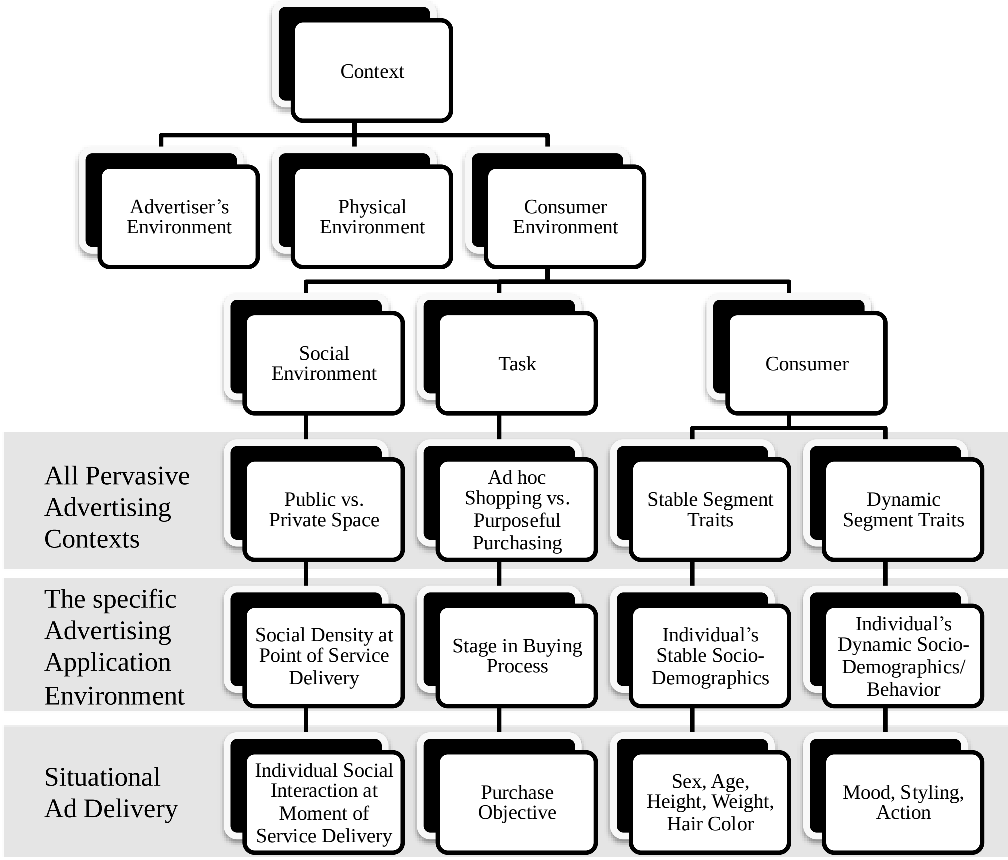

Fig. 5. Hierarchical structure on a macro, micro, and situational level [3]

5.5. Formative model evaluation

For evaluating and further developing the working model as part of the conceptualization, five academic experts were individually invited to serve as participants. On a plain wall, the working model was depicted with sticky notes, with each model item written on a single note. Additionally, the participants had a printout of the working model at their disposal. The participants were briefly informed about the context of the research, the first level of the model, and the systematics behind the macro, micro, and situational levels.

5.5.1. Top-down evaluation

The participants were first asked to interpret the overall working model while thinking-aloud so that their thoughts could be captured. They were also instructed to rearrange the sticky notes or introduce new ones until they were satisfied with the model. Interestingly, all participants – except one – first went through the entire model and only then started rearranging the notes.

Based on the insightful expert interviews, we rephrased many categories. For instance, we renamed manipulable and non-manipulable consumer profile to 'stable consumer profile' and 'situational consumer profile' because an interviewee said, "One can probably manipulate personality. Sex – even that is manipulable. Those are, let's say, at least long-term. This is nothing that one can change within the next five minutes. [...] There are states that may change dynamically and other that are rather long-term – rather stable." We omitted the category 'infrastructure', and moved 'design template' (formerly part of infrastructure) to the category 'location'. We integrated 'marketing policy' into 'advertising campaign', as suggested by the interviewees: "So, we have marketing policy, which is generally given. [...] And why is this part of infrastructure? Hmm... [...] Advertising campaign. Somehow this is also linked with that, isn't it? – Marketing."

5.5.2. Bottom-up evaluation

In the second exercise, participants were asked to classify a specific advertising adaptivity situation according to their rearranged working model. For this purpose, they were told to assume that a fully functional pervasive advertising system would be installed in a store. They should recall one of their last shopping experiences in a (physical) store and imagine that they encountered the context-adaptive advertising system. Again, they had the opportunity to rearrange the sticky notes or introduce new ones until they felt that they could accurately classify the situation. Interestingly, all participants felt satisfied with their models. Everybody could depict the situation in the model and none of them restructured it (again).

5.5.3. Iterations

As indicated, the result of this two-sided approach of card sorting was that we rephrased and rearranged many of the categories on all three (macro, micro, and situational) levels. After iterating the process with five experts, we achieved saturation, as no new insights popped up. Fig. 6 shows the resulting context model (taxonomy) for pervasive advertising.

5.6. Summative model evaluation

For the summative evaluation of the context model as depicted in **Error! Reference source not found.**, we opted for an assessment by the design team (i.e., human assessment). The design team used the following criteria for assessment: coverage of the advertising domain, applicability for the retailing environment (retail premises), and consistency.

However, the use of this specific context model (Fig. 6) in a real-world setting is beyond the scope of this paper. This model is to be regarded as an example result of the conceptualizing process. For practical implementation, more work must be done on how to systematically use the case study's context model in the targeted real-world setting (in this case, in retail premises).

6. Evaluation and Discussion

By using CCFIS, we conceptualized context for a specific intelligent, pervasive advertising system in retail. However, researchers/designers may find the system-specific context model that resulted from the case study too broad for implementing a fully functional system. Further narrowing the variety of context elements and highlighting their relationships within the envisaged system may be necessary.

Fig. 7 provides a simplified example of how the context model (Fig. 6) may be further particularized and transformed into an ontology for a specific contextual advertising scenario that could be

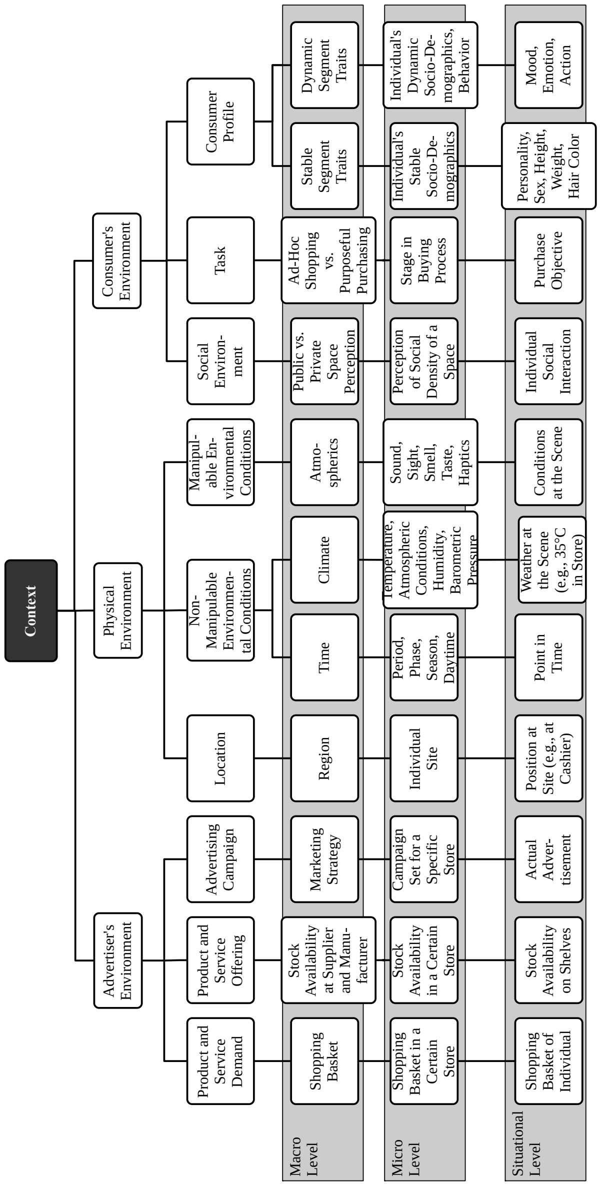

Fig. 6. A context model (taxonomy) for pervasive advertising in retail [3]

supported by the envisaged system. The shown ontology includes context elements that are more specific than the ones in the context model, and presents relationships within the envisaged scenario. By means of an excerpt of pseudo-code, Fig. 8 illustates how the ontology for the contextual advertising scenario (Fig. 7) may be transformed into opertional variables (e.g., rain=true). Note that it is just a simplified example for one fictitious scenario.

Still, this example argues in favor of the utility of the framework; it also pinpoints the next steps that have to be performed, which are outlined below.

First, researchers/designers must take additional steps (e.g., scenario method, low-fidelity

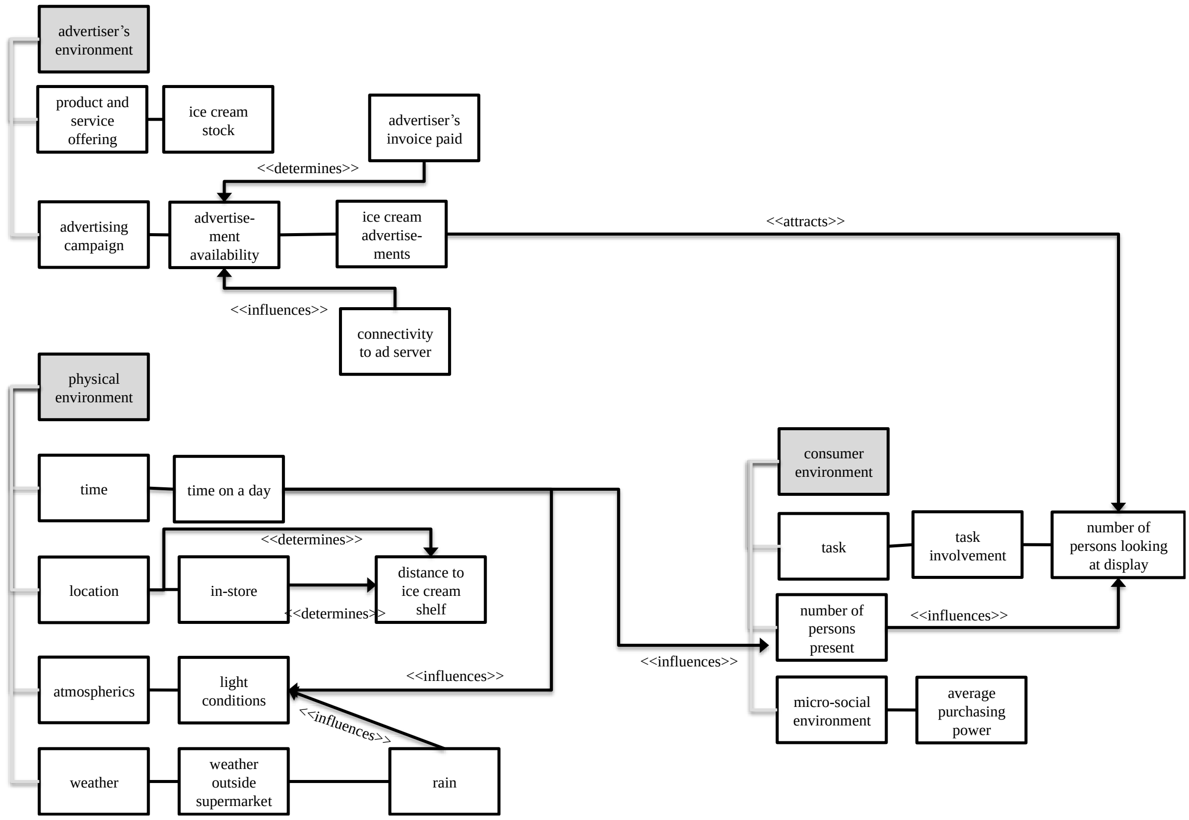

Fig. 7. Ontology for a contextual advertising scenario

prototyping) to select a limited set of context elements and specify the adaptivity logic that reveals the greatest utility for the system. Furthermore, the adaptivity logic (i.e., the interaction of context elements), which interrelates the selected context elements for the envisaged system, has to be defined. The interactions between context elements can be statically or dynamically operationalized. A static operationalization may use a deterministic logic, such as conditional statements (e.g., if-then-else constructs) [22]. An example of a dynamic operationalization would be one using artificial neural networks (e.g., such as those used for load management in power grids [28]).

Second, when developing an intelligent context-adaptive system, designers must consider the level of detail (e.g., granularity, required accuracy, or homogeneity) in which they need to specify context elements for a system to fulfill its purpose. Accordingly, they must use a technological approach. This task, though, is not within the scope of conceptualizing context.

When we compare the resulting system-specific context ontology of the case study with the presented generic context models (Tab. 1), we observe that the generic models' categories largely differ from those in the ontology. The ontology's categories are much more specific and particularly domain-specific. From this, we can conclude that the generic context models can, at most, serve as inspiration for the design of intelligent context-adaptive systems. CCFIS' process-oriented approach to context conceptualization appears to be superior for specifying the contextual requirements in real-world projects. Furthermore, the lack of domain-specific context aspects is a clear drawback of existing model-oriented approaches. Against this background, we reason that CCFIS better supports system designers in their task than static and purely generic approaches.

```
if (social_environment.number_of_persons_present >= 1
      && activity.task_involvement.number_of_persons_looking_at_display >= 1)
{
      if (social_environment.micro-social_environment.average_purchasing_power == "high")
            {set_producttype("luxury");}
      else {set_producttype("standard");}
      if (physical_environment.functional.instore == true
            && physical_environment.atmospherics.light_conditions == "dark")
            {set_advertisementcolors("bright");}
      else {set_advertisementcolors("standard");}

      if (physical_environment.weather.weather_outside_supermarket.rain == true)
            {show_advertisement ("umbrella");}
      } else {
            if (physical_environment.weather.weather_outside_supermarket == "hot"
                  && location.distance_to ice-cream shelf <= 3 meters
                  && advertisers_environment.product_offering.ice-cream_onstock == true)

                  if (advertiserX.last_invoice_paid == true
                        && resource_availability.advertiserX.ice-cream_onstock == true)
                        {show_advertisement ("ice-cream of advertiser X");}
            else {show_advertisement ("ice-cream of supermarket brand");}
      }
}

else {show_advertisement("supermarket self-advertisement");}
```

Fig. 8. A partial pseudo-code for a contextual advertising scenario

7. Conclusion

The key for any intelligent context-adaptive system is a thorough conceptualization of context. Existing approaches to conceptualize context represent an attempt to structure the variety of context in the form of generic context models. However, these models are static and do not target system design but are, in most cases, generated to define concepts; thus, these models support system designers only rudimentarily.

This paper proposes the framework 'CCFIS' for conceptualizing context to support system designers in designing intelligent context-adaptive systems. CCFIS follows a top-down and bottom-up approach and considers the perspectives of both system designer and user of an envisaged intelligent system. Furthermore, CCFIS integrates existing generic context models and builds on them. Moreover, CCFIS supports the opportunistic sensing paradigm [35], as it abstracts from sensors and allows for conceptualizing context on an implementation-independent level.

This work contributes to the context-aware research field in several ways. First, it provides designers of intelligent context-adaptive systems with a systematic framework for conceptualizing context, which is among the first that are based on a process methodology. The exemplary instantiation of CCFIS (case study) has demonstrated that it is applicable. Second, this work indicates the importance of considering domain-specific context, which we exemplified in the field of pervasive advertising.

A limitation of CCFIS is that researchers/designers cannot use it to identify an appropriate generic context model to build on. Yet, researchers are encouraged to come up with more supporting generic context models to build on and to propose applicable methods for model selection.

Furthermore, CCFIS is idealistic, assuming that any kind of context can be easily retrieved. However, while information on some context elements is technically easy to access (e.g., time or location), other elements – particularly those describing humans' inner states – may be challenging to gather (e.g., emotions, intentions). We, though, believe that context conceptualization and technical feasibility studies should be considered as separate phases in the design process. Thinking about technical challenges too early in the design process may result in less innovative solutions or even in systems that do not sufficiently reach the original adaptivity goals.

Finally, this work is limited as far as how applicable CCFIS is depending on the scale of a project, its resources (e.g., time, employees), and constraints (e.g., storage, performance). Also, this work does not cover the evaluation and quality assurance of ontologies derived by applying CCFIS. Future work should include a comparison of CCFIS to traditional, ad-hoc approaches and evaluate the results quantitatively and qualitatively.

CCFIS should inspire scholars to take a similar approach for conceptualizing context in various domains.

References

[1] S. S. Al-Fedaghi, Conceptualizing software life cycle, in *Information Systems: Modeling, Development, and Integration*, (2009), 438-457.

[2] C. Bauer, A Comparison and Validation of 13 Context Meta-Models, in *Proc. European Conference on Information Sytems (ECIS 2012)*, AIS (2012), Paper 17.

[3] C. Bauer and S. Spiekermann, Conceptualizing Context for Pervasive Advertising, in J. Müller, F. Alt and D. Michelis, eds., *Pervasive Advertising*, (2011), 159-183.

[4] B. Boehm, A spiral model of software development and enhancement, *ACM SIGSOFT Software Engineering Notes*, **11**, 4 (1986), 22-42.

[5] G. A. Bowen, Naturalistic inquiry and the saturation concept: a research note, *Qual Res*, **8**, 1 (February 2008), 137-152.

[6] N. A. Bradley and M. D. Dunlop, Toward a Multidisciplinary Model of Context to Support Context-Aware Computing, *Human-Computer Interaction*, **20**, 4 (2005), 403-446.

[7] P. J. Brown, J. D. Bovey and X. Chen, Context-aware applications: from the laboratory to the marketplace, *IEEE Personal Communications*, **4**, 5 (October 1997), 58-64.

[8] M. Broy and A. Rausch, Das neue V-Modell XT, *Informatik-Spektrum*, **28**, 3 (2005), 220-229.

[9] J. M. Carroll,*Scenario-based design: Envisioning work and technology in system development*, John Wiley and Sons, New York, NY, 1995.

[10] M. Chalmers, Awareness, Representation and Interpretation, *Computer Supported Cooperative Work*, **11**, 3-4 (2002), 389-409.

[11] M. Chalmers, A Historical View of Context, *Computer Supported Cooperative Work (CSCW)*, **13**, 3-4 (August 2004), 223-247.

[12] Y. Chen and M. E. Atwood, Context-Centered Design: Bridging the Gap Between Understanding and Designing, in J. A. Jacko, ed. *Human-Computer Interaction, Part 1 (HCII 2007)*, (2007), 40-48.

[13] J. Choi, Context-Driven Requirements Analysis, in *Proc. International Conference on Computational Science and its Applications (ICCSA 2007)*, Springer (2007).

[14] A. K. Dey and G. D. Abowd, Towards a better understanding of context and context-awareness, in *Proc. Workshop on The What, Who, Where, When, and How of Context-Awareness, part of CHI 2000* (2000).

[15] A. Dix, T. Rodden, N. Davies, J. Trevor, A. Friday and K. Palfreyman, Exploiting space and location as a design framework for interactive mobile systems, *ACM Transactions on Computer-Human Interaction*, **7**, 3 (September 2000), 285-321.

[16] S. Dobson, Leveraging the subtleties of location, in *Proc. sOc-EUSAI 2005*, ACM (2005), 189-193.

[17] A. Ferscha, C. Holzmann and S. Oppl, Context awareness for group interaction support, in *Proc. MobiWac 2004*, ACM (2004).

[18] A. Ferscha, S. Vogl and W. Beer, Ubiquitous context sensing in wireless environments, in P. Kacsuk, D. Kranzlmüller, Z. Németh and J. Volkert, eds., *Distributed and parallel systems: cluster and grid computing*, (2002).

[19] J. Floch, C. Frà, R. Fricke, K. Geihs, M. Wagner, J. Lorenzo, E. Soladana, S. Mehlhase, N. Paspallis, H. Rahnama, P. A. Ruiz and U. Scholz, Playing MUSIC: building context-aware and self-adaptive mobile applications, *Software: Practice and Experience*, **43**, 3 (March 2013), 359-388.

[20] A. Gómez-Pérez, Evaluation of Ontologies, *International Journal of Intelligent Systems*, **16**, 3 (2001), 391-409.

[21] A. Gómez-Pérez, A framework to verify knowledge sharing technology, *Expert Systems with Applications*, **11**, 4 (1996), 519-529.

[22] B. Guo, D. Zhang and M. Imai, Toward a cooperative programming framework for context-aware applications, *Personal and Ubiquitous Computing*, **15**, 3 (March 2011), 221-233.

[23] A. R. Hevner, S. T. March, J. Park and S. Ram, Design Science in Information Systems Research, *MIS Quarterly*, **28**, 1 (March 2004), 75-105.

[24] ISO 9241-11:1998(E), *Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11: Guidance on usability*, 1998.

[25] ISO 9241-210:2010 12 15, *Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems*, 2010.

[26] A. Jameson, Modelling both the Context and the User, *Personal and Ubiquitous Computing*, **5**, 1 (February 2001), 29-33.

[27] S. Jumisko-Pyykkö and T. Vainio, Framing the context of use for mobile HCI, *International Journal of Mobile Human Computer Interaction*, **2**, 4 (2010), 1-28.

[28] C. Köpp, H.-J. von Mettenheim and M. H. Breitner, Load Management in Power Grids, *Business & Information Systems Engineering*, **5**, 1 (2013), 35-44.

[29] I. Lee, J. Kim and J. Kim, Use Contexts for the Mobile Internet: A Longitudinal Study Monitoring Actual Use of Mobile Internet Services, *International Journal of Human-Computer Interaction*, **18**, 3 (2005), 269-292.

[30] H. Lieberman and T. Selker, Out of context: Computer systems that adapt to, and learn from, context, *IBM Systems Journal*, **39**, 3-4 (2000), 617-632.

[31] L. Lindstrom and R. Jeffries, Extreme Programming and Agile Software Development Methodologies, *Information Systems Management*, **21**, 3 (Summer 2004), 41-52.

[32] S. W. Loke, On representing situations for context-aware pervasive computing: six ways to tell if you are in a meeting, in *Proc. PerCom Workshops 2006*, IEEE (2006), 39-43.

[33] P. Lucas, Mobile Devices and Mobile Data: Issues of Identity and Reference, *Human-Computer Interaction*, **16**, 2-4 (2001), 323-336.

[34] P. Lukowicz, T. Choudhury and H. Gellersen, Beyond Context Awareness, *IEEE Pervasive Computing*, **10**, 4 (October-December 2011), 15-17.

[35] P. Lukowicz, S. Pentland and A. Ferscha, From Context Awareness to Socially Aware Computing, *IEEE Pervasive Computing*, **11**, 1 (January-March 2012), 32-40.

[36] W. Maass and S. Janzen, Pattern-Based Approach for Designing with Diagrammatic and Propositional Conceptual Models, in H. Jain, A. P. Sinha and P. Vitharana, eds., *DESRIST 2011*, *6629*, (2011), 192-206.

[37] S. Madsen and S. Matook, Conceptualizing Interpersonal Relationships in Agile IS Development, in *Proc. International Conference on Information Systems (ICIS 2010)*, AIS (2010).

[38] K. Maier and V. Stix, A semi-automated approach for structuring multi criteria decision problems, *European Journal of Operational Research*, **225**, 3 (March 2013), 487-496.

[39] L. Obrst, B. Ashpole, W. Ceusters, I. Mani, S. Ray and B. Smith, The Evaluation of Ontologies: Toward Improved Semantic Interoperability, in C. J. O. Baker and K.-H. Cheung, eds., *Semantic Web: Revolutionizing Knowledge Discovery in the Life Sciences*, (2007), 139-158.

[40] A. Oulasvirta, Finding meaningful uses for context-aware technologies: the humanistic research strategy, in *Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI 2004)* (2004), 247-254.

[41] J. Pascoe, Adding Generic Contextual Capabilities to Wearable Computers, in *Proc. 2nd International Symposium on Wearable Computers* (1998), 92-99.

[42] A. Pentland, Reality Mining of Mobile Communications: Toward a New Deal on Data, in H. Liu, J. Salerno and M. J. Young, eds., *Social Computing and Behavioral Modeling*, (2009), 75-80.

[43] P. Prekop and M. Burnett, Activities, context and ubiquitous computing, *Computer Communications*, **26**, 11 (July 2003), 1168-1176.

[44] W. W. Royce, Managing the development of large software systems, in *Proc. IEEE WESCON*, IEEE (1970), 1-9.

[45] J. Rubin and D. Chisnell,*Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*, Wiley, New York, NY, 2008.

[46] B. N. Schilit and M. M. Theimer, Disseminating Active Map Information to Mobile Hosts, *IEEE Network*, **8**, 5 (September-October 1994), 22-32.

[47] A. Schmidt, K. A. Aidoo, A. Takaluoma, U. Tuomela, K. Laerhoven and W. Velde, Advanced Interaction in Context, in H.-W. Gellersen, ed. *Handheld and Ubiquitous Computing*, *1707*, (1999), 89-101.

[48] A. Schmidt, M. Beigl and H.-W. Gellersen, There is more to Context than Location, *Computers & Graphics Journal*, **23**, 6 (December 1999), 893-901.

[49] E. Serral, P. Valderas and V. Pelechano, Towards the Model Driven Development of context-aware pervasive systems, *Pervasive and Mobile Computing* (2009).

[50] S. Sigg, S. Haseloff and K. David, An alignment approach for context prediction tasks in UbiComp environments, *IEEE Pervasive Computing*, **9**, 4 (October-December 2010), 90-97.

[51] B. Smith, From concepts to clinical reality: an essay on the benchmarking of biomedical terminologies, *Journal of Biomedical Informatics*, **39**, 3 (June 2006), 288-298.

[52] B. Smith and A. Kumar, Controlled vocabularies in bioinformatics: a case study in the gene ontology, *Drug Discovery Today: BIOSILICO*, **2**, 6 (2004), 246-252.

[53] P. Truillet, A Taxonomy of Physical Contextual Sensors, in J. A. Jacko, ed. *Human-Computer Interaction. Interaction Platforms and Techniques (Lecture Notes in Computer Science)*, *4551*, (2007), 982-989.

[54] O. Turel, Contextual Effects on the Usability Dimensions of Mobile Value-Added Services: a Conceptual Framework, *International Journal of Mobile Communications*, **4**, 3 (2006), 309-332.

[55] J. G. Walls, G. R. Widmeyer and O. A. El Sawy, Building an Information System Design Theory for Vigilant EIS, *Information Systems Research*, **3**, 1 (March 1992), 36-59.

[56] M. Weiser, The Computer for the 21st Century, *Scientific American*, **265**, 3 (September 1991), 94-104.

[57] Y. Yang, C. Onita, X. Zhang and J. Dhaliwal, TESTQUAL: conceptualizing software testing as a service, *e-Service Journal*, **7**, 2 (2010), 46-65.